Pastel (Home)

Standard for Interchange of USENET Messages

Mark R. Horton

[This memo is distributed as an RFC only to make this information easily accessible to researchers in the ARPA community. It does not specify an Internet standard.]

1. Introduction

This document defines the standard format for interchange of Network News articles among USENET sites. It describes the format for articles themselves, and gives partial standards for transmission of news. The news transmission is not entirely standardized in order to give a good deal of flexibility to the individual hosts to choose transmission hardware and software, whether to batch news, and so on.

There are five sections to this document. Section two section defines the format. Section three defines the valid control messages. Section four specifies some valid transmission methods. Section five describes the overall news propagation algorithm.

2. Article Format

The primary consideration in choosing an article format is that it fit in with existing tools as well as possible. Existing tools include both implementations of mail and news. (The notesfiles system from the University of Illinois is considered a news implementation.) A standard format for mail messages has existed for many years on the ARPANET, and this format meets most of the needs of USENET. Since the ARPANET format is extensible, extensions to meet the additional needs of USENET are easily made within the ARPANET standard. Therefore, the rule is adopted that all USENET news articles must be formatted as valid ARPANET mail messages, according to the ARPANET standard RFC 822. This standard is more restrictive than the ARPANET standard, placing additional requirements on each article and forbidding use of certain ARPANET features. However, it should always be possible to use a tool expecting an ARPANET message to process a news article. In any situation where this standard conflicts with the ARPANET standard, RFC 822 should be considered correct and this standard in error.

An example message is included to illustrate the fields.

```
Relay-Version: version B 2.10 2/13/83; site cbosgd.UUCP
Posting-Version: version B 2.10 2/13/83; site eagle.UUCP
Path: cbosgd!mhuxj!mhuxt!eagle!jerry
From: jerry@eagle.uucp (Jerry Schwarz)
Newsgroups: net.general
Subject: Usenet Etiquette -- Please Read
Message-ID: <642@eagle.UUCP>
Date: Friday, 19-Nov-82 16:14:55 EST
Followup-To: net.news
Expires: Saturday, 1-Jan-83 00:00:00 EST
Date-Received: Friday, 19-Nov-82 16:59:30 EST
Organization: Bell Labs, Murray Hill

The body of the article comes here, after a blank line.
```

Here is an example of a message in the old format (before the existence of this standard). It is recommended that implementations also accept articles in this format to ease upward conversion.

```
From: cbosgd!mhuxj!mhuxt!eagle!jerry (Jerry Schwarz)
Newsgroups: net.general
Title: Usenet Etiquette -- Please Read
Article-I.D.: eagle.642
Posted: Fri Nov 19 16:14:55 1982
Received: Fri Nov 19 16:59:30 1982
Expires: Mon Jan  1 00:00:00 1990

The body of the article comes here, after a blank line.
```

Some news systems transmit news in the "A" format, which looks like this:

```
Aeagle.642
net.general
cbosgd!mhuxj!mhuxt!eagle!jerry
Fri Nov 19 16:14:55 1982
Usenet Etiquette - Please Read
The body of the article comes here, with no blank line.
```

An article consists of several header lines, followed by a blank line, followed by the body of the message. The header lines consist of a keyword, a colon, a blank, and some additional information. This is a subset of the ARPANET standard, simplified to allow simpler software to handle it. The "from" line may optionally include a full name, in the format above, or use the ARPANET angle bracket syntax. To keep the implementations simple, other formats (for example, with part of the machine address after the close parenthesis) are not allowed. The ARPANET convention of continuation header lines (beginning with a blank or tab) is allowed.

Certain headers are required, certain headers are optional. Any unrecognized headers are allowed, and will be passed through unchanged. The required headers are Relay-Version, Posting-Version, From, Date, Newsgroups, Subject, Message-ID, Path. The optional headers are Followup-To, Date-Received, Expires, Reply-To, Sender, References, Control, Distribution, Organization.

2.1 Required Headers

2.1.1 Relay-Version This header line shows the version of the program responsible for the transmission of this article over the immediate link, that is, the program that is relaying the article from the next site. For example, suppose site A sends an article to site B, and site B forwards the article to site C. The message being transmitted from A to B would have a Relay-Version header identifying the program running on A, and the message transmitted from B to C would identify the program running on B. This header can be used to interpret older headers in an upward compatible way. Relay-Version must always be the first in a message; thus, all articles meeting this standard will begin with an upper case "R". No other restrictions are placed on the order of header lines.

The line contains two fields, separated by semicolons. The fields are the version and the full domain name of the site. The version should identify the system program used (e.g., "B") as well as a version number and version date. For example, the header line might contain

Relay-Version: version B 2.10 2/13/83; site cbosgd.UUCP

This header should not be passed on to additional sites. A relay program, when passing an article on, should include only its own Relay-Version, not the Relay-Version of some other site. (For upward compatibility with older software, if a Relay-Version is found in a header which is not the first line, it should be assumed to be moved by an older version of news and deleted.)

2.1.2 Posting-Version This header identifies the software responsible for entering this message into the network. It has the same format as Relay-Version. It will normally identify the same site as the Message-ID, unless the posting site is serving as a gateway for a message that already contains a message ID generated by mail. (While it is permissible for a gateway to use an externally generated message ID, the message ID should be checked to ensure it conforms to this standard and to RFC 822.)

2.1.3 From The From line contains the electronic mailing address of the person who sent the message, in the ARPA internet syntax. It may optionally also contain the full name of the person, in parentheses, after the electronic address. The electronic address is the same as the entity responsible for originating the article, unless the Sender header is present, in which case the From header might not be verified. Note that in all site and domain names, upper and lower case are considered the same, thus mark@cbosgd.UUCP, mark@cbosgd.uucp, and mark@CBosgD.UUcp are all equivalent. User names may or may not be case sensitive, for example, Billy@cbosgd.UUCP might be different from BillY@cbosgd.UUCP. Programs should avoid changing the case of electronic addresses when forwarding news or mail.

RFC 822 specifies that all text in parentheses is to be interpreted as a comment. It is common in ARPANET mail to place the full name of the user in a comment at the end of the From line. This standard specifies a more rigid syntax. The full name is not considered a comment, but an optional part of the header line. Either the full name is omitted, or it appears in parentheses after the electronic address of the person posting the article, or it appears before an electronic address enclosed in angle brackets. Thus, the three permissible forms are:

```
From: mark@cbosgd.UUCP
From: mark@cbosgd.UUCP (Mark Horton)
From: Mark Horton <mark@cbosgd.UUCP>
```

Full names may contain any printing ASCII characters from space through tilde, with the exceptions that they may not contain parentheses "(" or ")", or angle brackets "<" or ">". Additional restrictions may be placed on full names by the mail standard, in particular, the characters comma ",", colon ":", and semicolon ";" are inadvisable in full names.

2.1.4 Date The Date line (formerly "Posted") is the date, in a format that must be acceptable both to the ARPANET and to the getdate routine, that the article was originally posted to the network. This date remains unchanged as the article is propagated throughout the network. One format that is acceptable to both is

```
Weekday, DD-Mon-YY HH:MM:SS TIMEZONE
```

Several examples of valid dates appear in the sample article above. Note in particular that ctime format:

```
Wdy Mon DD HH:MM:SS YYYY
```

is not acceptable because it is not a valid ARPANET date. However, since older software still generates this format, news implementations are encouraged to accept this format and translate it into an acceptable format.

The contents of the TIMEZONE field is currently subject to worldwide time zone abbreviations, including the usual American zones (PST, PDT, MST, MDT, CST, CDT, EST, EDT), the other North American zones (Bering through Newfoundland), European zones, Australian zones, and so on. Lacking a complete list at present (and unsure if an unambiguous list exists), authors of software are encouraged to keep this code flexible, and in particular not to assume that time zone names are exactly three letters long. Implementations are free to edit this field, keeping the time the same, but changing the time zone (with an appropriate adjustment to the local time shown) to a known time zone.

2.1.5 Newsgroups The Newsgroups line specifies which newsgroup or newsgroups the article belongs in. Multiple newsgroups may be specified, separated by a comma. Newsgroups specified must all be the names of existing newsgroups, as no new newsgroups will be created by simply posting to them.

Wildcards (e.g., the word "all") are never allowed in a Newsgroups line. For example, a newsgroup "net.all" is illegal, although a newsgroup name "net.sport.football" is permitted.

If an article is received with a Newsgroups line listing some valid newsgroups and some invalid newsgroups, a site should not remove invalid newsgroups from the list. Instead, the invalid newsgroups should be ignored. For example, suppose site A subscribes to the classes "btl.all" and "net.all", and exchanges news articles with site B, which subscribes to "net.all" but not "btl.all". Suppose A receives an article with "Newsgroups: net.micro,btl.general". This article is passed on to B because B receives net.micro, but B does not receive btl.general. A must leave the Newsgroup line unchanged. If it were to remove "btl.general", the edited header could eventually reenter the "btl.all" class, resulting in an article that is not shown to users subscribing to "btl.general". Also, followups from outside "btl.all" would not be shown to such users.

2.1.6 Subject The Subject line (formerly "Title") tells what the article is about. It should be suggestive enough of the contents of the article to enable a reader to make a decision whether to read the article based on the subject alone. If the article is submitted in response to another article (e.g., is a "followup") the default subject should begin with the four characters "Re: " and the References line is required. (The user might wish to edit the subject of the followup, but the default should begin with "Re: ".)

2.1.7 Message-ID The Message-ID line gives the article a unique identifier. The same message ID may not be reused during the lifetime of any article with the same message ID. (It is recommended that no message ID be reused for at least two years.) Message ID's have the syntax

"<" "string not containing blank or >" ">"

In order to conform to RFC 822, the Message-ID must have the format

"<" "unique" "@" "full domain name" ">"

where "full domain name" is the full name of the host at which the article entered the network, including a domain that host is in, and unique is any string of printing ASCII characters, not including "<", ">", or "@". For example, the "unique" part could be an integer representing a sequence number for articles submitted to the network, or a short string derived from the date and time the article was created. For example, valid message ID for an article submitted from site ucbvax in domain Berkeley.ARPA would be "<4123@ucbvax.Berkeley.ARPA>". Programmers are urged not to make assumptions about the content of message ID fields from other hosts, but to treat them as unknown character strings. It is not safe, for example, to assume that a message ID will be under 14 characters, nor that it is unique in the first 14 characters.

The angle brackets are considered part of the message ID. Thus, in references to the message ID, such as the ihave/sendme and cancel control messages, the angle brackets are included. White space characters (e.g., blank and tab) are not allowed in a message ID. All characters between the angle brackets must be printing ASCII characters.

2.1.8 Path This line shows the path the article took to reach the current system. When a system forwards the message, it should add its own name to the list of systems in the Path line. The names may be separated by any punctuation character or characters, thus

"cbosgd!mhuxj!mhuxt", "cbosgd, mhuxj, mhuxt", and "@cbosgd.uucp,@mhuxj.uucp,@mhuxt.uucp" and even "teklabs, zehntel, sri-unix@cca!decvax" are valid entries. (The latter path indicates a message that passed through decvax, cca, sri-unix, zehntel, and teklabs, in that order.) Additional names should be added from the left, for example, the most recently added name in the third example was "teklabs". Letters, digits, periods and hyphens are considered part of site names; other punctuation, including blanks, are considered separators.

Normally, the rightmost name will be the name of the originating system. However, it is also permissible to include an extra entry on the right, which is the name of the sender. This is for upward compatibility with older system.

The Path line is not used for replies, and should not be taken as a mailing address. It is intended to show the route the message travelled to reach the local site. There are several uses for this information. One is to monitor USENET routing for performance reasons. Another is to establish a path to reach new sites. Perhaps the most important is to cut down on redundant USENET traffic by failing to forward a message to a site that is known to have already received it. In particular, when site A sends an article to site B, the Path line includes "A", so that site B will not immediately send the article back to site A. The site name each site uses to identify itself should be the same as the name by which its neighbors know it, in order to make this optimization possible.

A site adds its own name to the front of a path when it receives a message from another site. Thus, if a message with path A!X!Y!Z is passed from site A to site B, B will add its own name to the path when it receives the message from A, e.g., B!A!X!Y!Z. If B then passes the message on to C, the message sent to C will contain the path B!A!X!Y!Z, and when C receives it, C will change it to C!B!A!X!Y!Z.

Special upward compatibility note: Since the From, Sender, and Reply-To lines are in internet format, and since many USENET sites do not yet have mailers capable of understanding internet format, it would break the reply capability to completely sever the connection between the Path header and the reply function. Thus, sites are required to continue to keep the Path line in a working reply format as much as possible, until January 1, 1984. It is recognized that the path is not always a valid reply string in older implementations, and no requirement to fix this problem is placed on implementations. However, the

existing convention of placing the site name and an "!" at the front of the path, and of starting the path with the site name, an "!", and the user name, should be maintained at least until 1984.

2.2 Optional Headers

2.2.1 Reply-To This line has the same format as From. If present, mailed replies to the author should be sent to the name given here. Otherwise, replies are mailed to the name on the From line. (This does not prevent additional copies from being sent to recipients named by the replier, or on To or Cc lines.) The full name may be optionally given, in parentheses, as in the From line.

2.2.2 Sender This field is present only if the submitter manually enters a From line. It is intended to record the entity responsible for submitting the article to the network, and should be verified by the software at the submitting site.

For example, if John Smith is visiting CCA and wishes to post an article to the network, using friend Sarah Jones account, the message might read

```
From: smith@ucbvax.uucp (John Smith)
Sender: jones@cca.arpa (Sarah Jones)
```

If a gateway program enters a mail message into the network at site sri-unix, the lines might read

```
From: John.Doe@CMU-CS-A.ARPA
Sender: network@sri-unix.ARPA
```

The primary purpose of this field is to be able to track down articles to determine how they were entered into the network. The full name may be optionally given, in parentheses, as in the From line.

2.2.3 Followup-To This line has the same format as Newsgroups. If present, follow-up articles are to be posted to the newsgroup(s) listed here. If this line is not present, followups are posted to the newsgroup(s) listed in the Newsgroups line, except that followups to "net.general" should instead go to "net.followup".

2.2.4 Date-Received This line (formerly "Received") is in a legal USENET date format. It records the date and time that the article was first received on the local system. If this line is present in an article being transmitted from one host to another, the receiving host should ignore it and replace it with the current date. Since this field is intended for local use only, no site is required to support it. However, no site should pass this field on to another site unchanged.

2.2.5 Expires This line, if present, is in a legal USENET date format. It specifies a suggested expiration date for the article. If not present, the local default expiration date is used.

This field is intended to be used to clean up articles with a limited usefulness, or to keep important articles around for longer than usual. For example, a message announcing an upcoming seminar could have an expiration date the day after the seminar, since the message is not useful after the seminar is over. Since local sites have local policies for expiration of news (depending on available disk space, for instance), users are discouraged from providing expiration dates for articles unless there is a natural expiration date associated with the topic. System software should almost never provide a default Expires line. Leave it out and allow local policies to be used unless there is a good reason not to.

2.2.6 References This field lists the message ID's of any articles prompting the submission of this article. It is required for all follow-up articles, and forbidden when a new subject is raised. Implementations should provide a follow-up command, which allows a user to post a follow-up article. This command should generate a Subject line which is the same as the original article, except that if the original subject does not begin with "Re: " or "re: ", the four characters "Re: " are inserted before the subject. If there is no References line on the original header, the References line snould contain the message ID of the original article (including the angle brackets). If the original article does have a References line, the followup article should have a References line containing the text of the original References line, a blank, and the message ID of the original article.

The purpose of the References header is to allow articles to be grouped into conversations by the user interface program. This allows conversations within a newsgroup to be kept together, and potentially users might shut off entire conversations without unsubscribing to a newsgroup. User interfaces may not make use of this header, but all automatically generated followups should generate the References line for the benefit of systems that do use it, and manually generated followups (e.g. typed in well after the original article has been printed by the machine) should be encouraged to include them as well.

2.2.7 Control If an article contains a Control line, the article is a control message. Control messages are used for communication among USENET host machines, not to be read by users. Control messages are distributed by the same newsgroup mechanism as ordinary messages. The body of the Control header line is the message to the host.

For upward compatibility, messages that match the newsgroup pattern "all.all.ctl" should also be interpreted as control messages. If no Control: header is present on such messages, the subject is used as the control message. However, messages on newsgroups matching this pattern do not conform to this standard.

2.2.8 Distribution This line is used to alter the distribution scope of the message. It has the same format as the Newsgroups line. User subscriptions are still controlled by Newsgroups, but the message is sent to all systems subscribing to the newsgroups on the Distribution line instead of the Newsgroups line. Thus, a car for sale in New Jersey might have headers including

```
Newsgroups: net.auto,net.wanted
Distribution: nj.all
```

so that it would only go to persons subscribing to net.auto or net.wanted within New Jersey. The intent of this header is to further restrict the distribution of a newsgroup, not to increase it. A local newsgroup, such as nj.crazy-eddie, will probably not be propagated by sites outside New Jersey that do not show such a newsgroup as valid. Wildcards in newsgroup names in the Distribution line are allowed. Followup articles should default to the same Distribution line as the original article, but the user can change it to a more limited one, or escalate the distribution if it was originally restricted and a more widely distributed reply is appropriate.

2.2.9 Organization The text of this line is a short phrase describing the organization to which the sender belongs, or to which the machine belongs. The intent of this line is to help identify the person posting the message, since site names are often cryptic enough to make it hard to recognize the organization by the electronic address.

3. Control Messages

This section lists the control messages currently defined. The body of the Control header is the control message. Messages are a sequence of zero or more words, separated by white space (blanks or tabs). The first word is the name of the control message, remaining words are parameters to the message. The remainder of the header and the body of the message are also potential parameters; for example, the From line might suggest an address to which a response is to be mailed.

Implementors and administrators may choose to allow control messages to be automatically carried out, or to queue them for manual processing. However, manually processed messages should be dealt with promptly.

3.1 Cancel

cancel <message ID>

If an article with the given message ID is present on the local system, the article is cancelled. This mechanism allows a user to cancel an article after the article has been distributed over the network.

Only the author of the article or the local super user is allowed to use this message. The verified sender of a message is the Sender line, or if no Sender line is present, the From line. The verified sender of the cancel message must be the same as either the Sender or From field of the original message. A verified sender in the cancel message is allowed to match an unverified From in the original message.

3.2 Ihave/Sendme

ihave <message ID list> <remotesys>
sendme <message ID list> <remotesys>

This message is part of the "ihave/sendme" protocol, which allows one site (say "A") to tell another site ("B") that a particular message has been received on A. Suppose that site A receives article "ucbvax.1234", and wishes to transmit the article to site B. A sends the control message "ihave ucbvax.1234 A" to site B (by posting it to newsgroup "to.B"). B responds with the control message "sendme ucbvax.1234 B" (on newsgroup to.A) if it has not already received the article. Upon receiving the Sendme message, A sends the article to B.

This protocol can be used to cut down on redundant traffic between sites. It is optional and should be used only if the particular situation makes it worthwhile. Frequently, the outcome is that, since most original messages are short, and since there is a high overhead to start sending a new message with UUCP, it costs as much to send the Ihave as it would cost to send the article itself.

One possible solution to this overhead problem is to batch requests. Several message ID's may be announced or requested in one message. If no message ID's are listed in the control message, the body of the message should be scanned for message ID's, one per line.

3.3 Newgroup

newgroup <groupname>

This control message creates a new newsgroup with the name given. Since no articles may be posted or forwarded until a newsgroup is created, this message is required before a newsgroup can be used. The body of the message is expected to be a short paragraph describing the intended use of the newsgroup.

3.4 Rmgroup

rmgroup <groupname>

This message removes a newsgroup with the given name. Since the newsgroup is removed from every site on the network, this command should be used carefully by a responsible administrator.

3.5 Sendsys

sendsys (no arguments)

The "sys" file, listing all neighbors and which newsgroups are sent to each neighbor, will be mailed to the author of the control message (Reply-to, if present, otherwise From). This information is considered public information, and it is a requirement of membership in USENET that this information be provided on request, either automatically in response to this control message, or manually, by mailing the requested information to the author of the message. This information is used to keep the map of USENET up to date, and to determine where netnews is sent.

The format of the file mailed back to the author should be the same as that of the "sys" file. This format has one line per neighboring site (plus one line for the local site), containing four colon separated fields. The first field has the site name of the neighbor, the second field has a newsgroup pattern describing the newsgroups sent to the neighbor. The third and fourth fields are not defined by this standard. A sample response:

```
From cbosgd!mark  Sun Mar 27 20:39:37 1983
Subject: response to your sendsys request
To: mark@cbosgd.UUCP
```

```
Responding-System: cbosgd.UUCP
cbosgd:osg,cb,btl,bell,net,fa,to,test
ucbvax:net,fa,to.ucbvax:L:
cbosg:net,fa,bell,btl,cb,osg,to.cbosg:F:/usr/spool/outnews/cbosg
cbosgb:osg,to.cbosgb:F:/usr/spool/outnews/cbosgb
sescent:net,fa,bell,btl,cb,to.sescent:F:/usr/spool/outnews/sescent
npois:net,fa,bell,btl,ug,to.npois:F:/usr/spool/outnews/npois
mhuxi:net,fa,bell,btl,ug,to.mhuxi:F:/usr/spool/outnews/mhuxi
```

3.6 Senduuname

```
senduuname      (no arguments)
```

The "uuname" program is run, and the output is mailed to the author of the control message (Reply-to, if present, otherwise From). This program lists all uucp neighbors of the local site. This information is used to make maps of the UUCP network. The sys file is not the same as the UUCP L.sys file. The L.sys file should never be transmitted to another party without the consent of the sites whose passwords are listed therein.

It is optional for a site to provide this information. Some reply should be made to the author of the control message, so that a transmission error won't be blamed. It is also permissible for a site to run the uuname program (or in some other way determine the uucp neighbors) and edit the output, either automatically or manually, before mailing the reply back to the author. The file should contain one site per line, beginning with the uucp site name. Additional information may be included, separated from the site name by a blank or tab. The phone number or password for the site should NOT be included, as the reply is considered to be in the public domain. (The uuname program will send only the site name and not the entire contents of the L.sys file, thus, phone numbers and passwords are not transmitted.)

The purpose of this message is to generate and maintain UUCP mail routing maps. Thus, connections over which mail can be sent using the site!user syntax should be included, regardless of whether the link is actually a UUCP link at the physical level. If a mail router should use it, it should be included. Since all information sent in response to this message is optional, sites are free to edit the list, deleting secret or private links they do not wish to publicise.

3.7 Version

```
version (no arguments)
```

The name and version of the software running on the local system is to be mailed back to the author of the article (Reply-to if present, otherwise From).

4. Transmission Methods

USENET is not a physical network, but rather a logical network resting on top of several existing physical networks. These networks include, but are not limited to, UUCP, the ARPANET, an Ethernet, the BLICN network, an NSC Hyperchannel, and a Berknet. What is important is that two neighboring systems on USENET have some method to get a new article, in the format listed here, from one system to the other, and once on the receiving system, processed by the netnews software on that system. (On UNIX systems, this usually means the "rnews" program being run with the article on the standard input.)

It is not a requirement that USENET sites have mail systems capable of understanding the ARPA Internet mail syntax, but it is strongly recommended. Since From, Reply-To, and Sender lines use the Internet syntax, replies will be difficult or impossible without an internet mailer. A site without an internet mailer can attempt to use the Path header line for replies, but this field is not guaranteed to be a working path for replies. In any event, any site generating or forwarding news messages must have an internet address that allows them to receive mail from sites with internet mailers, and they must include their internet address on their From line.

4.1 Remote Execution

Some networks permit direct remote command execution. On these networks, news may be forwarded by spooling the rnews command with the article on the standard input. For example, if the remote system is called "remote", news would be sent over a UUCP link with the command "uux - remote!rnews", and on a Berknet, "net -mremote rnews". It is important that the article be sent via a reliable mechansim, normally involving the possibility of spooling, rather than direct real-time remote execution. This is because, if the remote system is down, a direct execution command will fail, and the article will never be delivered. If the article is spooled, it will eventually be delivered when both systems are up.

4.2 Transfer by Mail

On some systems, direct remote spooled execution is not possible. However, most systems support electronic mail, and a news article can be sent as mail. One approach is to send a mail message which is identical to the news message: the mail headers are the news headers, and the mail body is the news body. By convention, this mail is sent to the user "newsmail" on the remote machine.

One problem with this method is that it may not be possible to convince the mail system that the From line of the message is valid, since the mail message was generated by a program on a system different from the source of the news article. Another problem is that error messages caused by the mail transmission would be sent to the originator of the news article, who has no control over news transmission between two cooperating hosts and does not know who to contact. Transmission error messages should be directed to a responsible contact person on the sending machine.

A solution to this problem is to encapsulate the news article into a mail message, such that the entire article (headers and body) are part of the body of the mail message. The convention here is that such mail is sent to user "rnews" on the remote system. A mail message body is generated by prepending the letter "N" to each line of the news article, and then attaching whatever mail headers are convenient to generate. The N's are attached to prevent any special lines in the news article from interfering with mail transmission, and to prevent any extra lines inserted by the mailer (headers, blank lines, etc.) from becoming part of the news article. A program on the receiving machine receives mail to "rnews", extracting the article itself and invoking the "rnews" program. An example in this format might look like this:

```
Date: Monday, 3-Jan-83 08:33:47 MST
From: news@cbosgd.UUCP
Subject: network news article
To: rnews@npois.UUCP

NRelay-Version: B 2.10  2/13/83 cbosgd.UUCP
NPosting-Version: B 2.9 6/21/82 sask.UUCP
NPath: cbosgd!mhuxj!harpo!utah-cs!sask!derek
NFrom: derek@sask.UUCP (Derek Andrew)
NNewsgroups: net.test
NSubject: necessary test
NMessage-ID: <176@sask.UUCP>
NDate: Monday, 3-Jan-83 00:59:15 MST
N
NThis really is a test.  If anyone out there more than 6
Nhops away would kindly confirm this note I would
Nappreciate it.  We suspect that our news postings
Nare not getting out into the world.
N
```

Using mail solves the spooling problem, since mail must always be spooled if the destination host is down. However, it adds more overhead to the transmission process (to encapsulate and extract the article) and makes it harder for software to give different priorities to news and mail.

4.3 Batching

Since news articles are usually short, and since a large number of messages are often sent between two sites in a day, it may make sense to batch news articles. Several articles can be combined into one large article, using conventions agreed upon in advance by the two sites. One such batching scheme is described here; its use is still considered experimental.

News articles are combined into a script, separated by a header of the form:

```
##! rnews 1234
```

where 1234 is the length, in bytes, of the article. Each such line is followed by an article containing the given number of bytes. (The newline at the end of each line of the article is counted as one byte, for purposes of this count, even if it is stored as CRLF.) For example, a batch of articles might look like this:

```
#! rnews 374
Relay-Version: version B 2.10 2/13/83; site cbosgd.UUCP
Posting-Version: version B 2.10 2/13/83; site eagle.UUCP
Path: cbosgd!mhuxj!mhuxt!eagle!jerry
From: jerry@eagle.uucp (Jerry Schwarz)
Newsgroups: net.general
Subject: Usenet Etiquette -- Please Read
Message-ID: <642@eagle.UUCP>
Date: Friday, 19-Nov-82 16:14:55 EST

Here is an important message about USENET Etiquette.
#! rnews 378
Relay-Version: version B 2.10 2/13/83; site cbosgd.UUCP
Posting-Version: version B 2.10 2/13/83; site eagle.UUCP
Path: cbosgd!mhuxj!mhuxt!eagle!jerry
From: jerry@eagle.uucp (Jerry Schwarz)
Newsgroups: net.followup
Subject: Notes on Etiquette article
Message-ID: <643@eagle.UUCP>
Date: Friday, 19-Nov-82 17:24:12 EST

There was something I forgot to mention in the last message.
```

Batched news is recognized because the first character in the message is "#". The message is then passed to the unbatcher for interpretation.

5. The News Propagation Algorithm

This section describes the overall scheme of USENET and the algorithm followed by sites in propagating news to the entire network. Since all sites are affected by incorrectly formatted articles and by propagation errors, it is important for the method to be standardized.

USENET is a directed graph. Each node in the graph is a host computer, each arc in the graph is a transmission path from one host to another host. Each arc is labelled with a newsgroup pattern, specifying which newsgroup classes are forwarded along that link. Most arcs are bidirectional, that is, if site A sends a class of newsgroups to site B, then site B usually sends the same class of newsgroups to site A. This bidirectionality is not, however, required.

USENET is made up of many subnetworks. Each subnet has a name, such as "net" or "btl". The special subnet "net" is defined to be USENET, although the union of all subnets may be a superset of USENET (because of sites that get local newsgroup classes but do not get net.all). Each subnet is a connected graph, that is, a path exists from every node to every other node in the subnet. In addition, the entire graph is (theoretically) connected. (In practice, some political considerations have caused some sites to be unable to post articles reaching the rest of the network.)

An article is posted on one machine to a list of newsgroups. That machine accepts it locally, then forwards it to all its neighbors that are interested in at least one of the newsgroups of the message. (Site A deems site B to be "interested" in a newsgroup if the newsgroup matches the pattern on the arc from A to B. This pattern is stored in a file on the A machine.) The sites receiving the incoming article examine it to make sure they really want the article, accept it locally, and then in turn forward the article to all their interest neighbors. This process continues until the entire network has seen the article.

An important part of the algorithm is the prevention of loops. The above process would cause a message to loop along a cycle forever. In particular, when site A sends an article to site B, site B will send it back to site A, which will send it to site B, and so on. One solution to this is the history mechanism. Each site keeps track of all articles it has seen (by their message ID) and whenever an article comes in that it has already seen, the incoming article is discarded immediately. This solution is sufficient to prevent loops, but additional optimizations can be made to avoid sending articles to sites that will simply throw them away.

One optimization is that an article should never be sent to a machine listed in the Path line of the header. When a machine name is in the Path line, the message is known to have passed through the machine. Another optimization is that, if the article originated on site A, then site A has already seen the article. (Origination can be determined by the Posting-Version line.)

Thus, if an article is posted to newsgroup "net.misc", it will match the pattern "net.all" (where "all" is a metasymbol that matches any string), and will be forwarded to all sites that subscribe to net.all (as determined by what their neighbors send them). These sites make up the "net" subnetwork. An article posted to "btl.general" will reach all sites receiving "btl.all", but will not reach sites that do not get "btl.all". In effect, the articles reaches the "btl" subnetwork. An article posted to newsgroups "net.micro,btl.general" will reach all sites subscribing to either of the two classes.